On Locally Linear Classification by Pairwise Coupling

Feng Chen, Chang-Tien Lu, Arnold P. Boedihardjo
Virginia Polytechnic Institute and State University
7054 Haycock Road, Falls Church, VA, 22043
{chenf, ctlu, arnold.p.boedihardjo}@vt.edu

Abstract

Locally linear classification by pairwise coupling addresses a nonlinear classification problem by three basic phases: decompose the classes of complex concepts into linearly separable subclasses, learn a linear classifier for each pair, and combine pairwise classifiers into a single classifier. A number of methods have been proposed in this framework. However, these methods have several deficiencies: 1) lack of a systematic evaluation of this framework, 2) naive application of general clustering algorithms to generate subclasses, and 3) no valid method to estimate an optimal number of subclasses. This paper proves the equivalence between three popular combination schemas under general settings, defines several global criterion functions for measuring the goodness of subclasses, and presents a supervised greedy clustering algorithm to minimize the proposed criterion functions. Extensive experiments has also been conducted on a set of benchmark data to validate the effectiveness of the proposed techniques.

1 Introduction

In recent years, there has been an emerging interest to solve a complex (nonlinear) classification problem by using locally linear classification (LLC) techniques [1–6]. The basic idea is to approximate a nonlinear decision boundary by consecutive segments, each of which is determined by a local linear classifier. A nonlinear classification problem can be decomposed into a series of linear classification subtasks, such that the problem can be solved linearly in the original space. Results have shown that this approach can achieve competitive generalization accuracy and higher training efficiency than other advanced approaches such as neural network [3], generalized linear discriminative analysis [1, 5], and nonlinear support vector machines [9, 17].

The effectiveness of LLC lies in the fact that each local classifier requires estimating a much simpler target function, thus reducing the chance of overfitting. However, as a potential disadvantage, more target functions need to be estimated with less training data. An implicit assumption of LLC is that the gain acquired by the reduced complexity is more than the loss incurred by the "reduced" training data. LLC includes three major categories: pairwise coupling based (LLC-PC) [2, 3, 6, 8], local space based (LLC-LS) [4], and model based (LLC-MD) [1, 5, 13]. LLC-PC decomposes the classes of complex concepts into linearly separable subclasses, then learns a linear prototype classifier for each pair of subclasses, and finally combines the pairwise prototype classifiers into a single classifier. LLC-LS divides the input space into several disjoint subspaces, and then learns a linear classifier for each subspace. LLC-MD assumes each class as a mixture of normals and learns a linear discriminant analysis (LDA) classifier by treating each normal as a pseudo-class.

This paper focuses on LLC-PC, the **L**ocally **L**inear **C**lassification by **P**airwise **C**oupling. It is a natural generalization of the state-of-the-art multiclass classification approach by pairwise coupling [10, 11]. As to the major extension, LLC-PC differentiates the pairs of subclasses with the same parent class label from those with different parent class labels. Existing methods for LLC-PC apply naive general clustering methods (e.g., k-means) to generate subclasses, and empoly different combination schemas (e.g., voting, MinMax) to integrate pairwise prototype classifiers [2, 3, 6, 8]. Some empirical comparisons demonstrate the similar classification accuracy between different combination schemas [3, 8]. However, there is no research presented to explain this phenomenon. As shown later, the explanation will lead to a new reformulation of the pairwise coupling problem as a voronoi diagram problem, thus introducing a new direction to further optimize LLC-PC.

In this paper, we address the following issues: First, the generation of appropriate subclasses can not be optimally solved by directly applying general clustering algorithms. This is due to the main principle for solving problems using a restricted amount of information: "When solving a given problem, try to avoid solving a more general problem as an intermediate step [16]." A supervised clustering algorithm must be redesigned by considering the impacts of

other phases. Second, there should exist some connections between different combination schemas, in order to explain the fact that they usually exhibit similar classification accuracy. Third, the existing methods require users to predefine the number of subclasses per-class. However, it is difficult to determine an optimal number of subclasses, such that the predictive accuracy of the resulting classifier can be maximized. Our major contributions can be summarized as follows:

- Prove the weak equivalence between three popular combination schemas on the generalization accuracy;
- Demonstrate the tradeoff correlation between the cluster granularity and the generalization accuracy of the resulting classifier, similar to the Bias-Variance dilemma [17];
- Define several global criterion functions and evaluate their major characteristics, such as the monotonicity and computational cost;
- Propose an effective greedy algorithm to identify good-quality subclasses, and present a method to estimate the optimal number of subclasses for different subclasses generation methods.

The rest of the paper is organized as follows. Section 2 reviews three categories of related methods. Section 3 introduces preliminaries of LLC-PC, including the combination schemas and the correlation between cluster granularity and generalization accuracy. Section 4 defines new criterion functions and discuses their major characteristics. A greedy subclasses generation algorithm is presented in Section 5. Section 6 demonstrates experiment results on benchmark data sets. Finally, we conclude with future work in Section 7.

2 Related Work

This section summarizes three major categories of works on LLC, including pairwise coupling, local space, and model based approaches.

Pairwise Coupling Based (LLC-PC). Schulmeister *et al*. presented a method named hybrid piecewise linear classifier (DIPOL), which first uses k-means to generate subclasses, then applies a linear Perceptron model to build pairwise prototype classifiers, and finally applies a voting schema to perform combination [2]. This approach was later extended to support vector machines by replacing the Perceptron model with a linear SVM model [7]. Lu *et al*. proposed a similar method for massively parallel training of neural networks [3]. They used random and grid-space partition algorithms as the clustering algorithm and employed a new combination schema, named min-max module, to integrate the pairwise prototype classifiers. This method was later extended to support vector machines by using a linear SVM model to build prototype classifiers [8]. Wu *et al*. proposed the general framework of LLC-PC and applied it to the classification of rare classes [6].

Local Space Based (LLC-LS). Kim *et al*. proposed an approach called locally linear discriminant analysis (LLDA), which decomposes the training set into several local clusters by using k-means, and then learns a LDA classifier for each local cluster [4]. Cheng *et al*. presented a method named localized support vector machines, which decomposes the training data into local clusters by using a proposed supervised k-means algorithm (MagKmeans), and then learns a support vector machine for each local cluster [9].

Model Based (LLC-MD). Hastie *et al*. presented a method named mixture discriminative analysis (MDA), which uses a supervised EM algorithm to identify subclasses and a linear discriminant analysis (LDA) model to build the classifier by regarding each subclass as a pseudo-class [1]. Later, several improvements were proposed to alleviate the constraints of the LDA model [13, 14]. Zhu *et al*. presented several criterions to estimate the optimal number of subclasses (Gaussians) [5].

Differences and Advantages of LLC-PC: The major difference between LLC-PC and LLC-LS is that LLC-PC conducts a clustering process in each class to generate subclasses, whereas LLC-LS conducts only one clustering process over all the training data, without differentiating the instances of different class labels. Given a new object $\mathbf{x}$, LLC-PC applies pairwise classifiers to $\mathbf{x}$ and then ensembles the results to classify $\mathbf{x}$, whereas LLC-LS identifies the cluster closest to $\mathbf{x}$, and then applies the corresponding classifier to $\mathbf{x}$.

LLC-PC has some distinct advantages over the others. First, LLC-LS has the restriction that each subclass only contributes to a single local classifier, whereas in LLC-PC each sub-class can contribute to multiple local classifiers. Figure 1 shows a data set of the classic XOR problem. There are two classes $\{C_1, C_2\}$, and suppose C_i is a mixture of two normals $\{C_{i1}, C_{i2}\}$, where $i = 1..2$. Clearly, each normal contributes to two local classifiers. For instance, C_{11} contributes to the local classifiers between C_{11} and C_{22}, and between C_{11} and C_{21}. Unlike LLC-PC, it is difficult to apply LLC-LS to this problem. A possible solution for LLC-LS is to divide each normal into two subclasses, but it will unnecessarily reduce the sample size of each subclass by half. Second, because of the Gaussian assumption, LLC-MD may not perform well for non-Gaussian data [13]. Furthermore, LLC-MD only learns a single LDA classifier for all subclasses. In comparison, LLC-PC is useful for both Gaussian and non-Gaussian data. It supports class specific feature extraction and classifier, and each local classifier can be regarded as a completely separate classification problem and can be

learned in parallel.

3 Preliminaries

This section discusses two basic components of LLC-PC, including cluster granularity (e.g., the total number of clusters generated) and combination schema. We introduce three popular combination schemas, prove their weak equivalence on classification accuracy, and then present the correlation between cluster granularity and classification accuracy.

3.1 Combination Schemas

There are three popular strategies that can be used to combine pairwise prototype classifiers into a single classifier, namely, the voting-based [2, 6], the probability-based [11], and the minimum and maximization principles based [3] (abbreviated as MinMax). Suppose there are N classes $\{C_1, C_2, \ldots, C_N\}$, each class C_i $(i = 1, \ldots, N)$ is divided into N_i pseudo clusters $(C_{i1}, C_{i2}, \ldots, C_{iN_i})$, and the separating hyperplane for C_{ij} and C_{kp} is $f_{ij-kp}(\mathbf{x}) = \mathbf{w}_{ij-kp}^T \mathbf{x} + b$. The preceding combination schemas can be summarized as follows:

Voting based: The decision function for the subclass C_{ij} can be defined by $F_{ij}(\mathbf{x}) = \sum_{k \neq i, k=1}^{N} \sum_{p=1}^{N_k} (\delta(f_{ij-kp}(\mathbf{x})))$, where $\delta(z) = 1$ if $z \geq 0$, and 0 otherwise. The decision function for the class C_i can be defined by $F_i(\mathbf{x}) = max(F_{ij}(\mathbf{x}) / \sum_{o=1, o \neq i}^{N} N_o)$, where $1 \leq j \leq N_i$, and the denominator is used for normalization, since the number of subclasses generated for each class may be different. The new point $\mathbf{x}$ is classified as follows: $G(\mathbf{x}) = \underset{i=1,\ldots,N}{arg\,max}(F_i(\mathbf{x}))$. If a new object exists in an unclassified region, the object is classified on the basis of the minimum distance to the class regions.

Probability based: The decision function $F_{ij}(\mathbf{x})$ can be defined by $F_{ij}(\mathbf{x}) = Prob(y = C_{ij}|\mathbf{x})$, where the posterior probability $Prob(y = C_{ij}|\mathbf{x})$ can be estimated from the available pairwise class probabilities $Prob_{ij-kp} = Prob(y = C_{ij}|y = C_{ij}\ or\ C_{kp}, \mathbf{x})$ [11]. The decision function $F_i(\mathbf{x})$ is defined by $F_i(\mathbf{x}) = max(F_{ij}(\mathbf{x}))$, where $1 \leq j \leq N_i$. The new point x is classified as follows: $G(\mathbf{x}) = \underset{i=1,\ldots,N}{argmax}(F_i(\mathbf{x}))$.

MinMax based: The decision function $F_{ij}(\mathbf{x})$ can be defined by $F_{ij}(\mathbf{x}) = min(f_{ij-kp}(\mathbf{x}))$, where $k \neq i$. The decision function $F_i(\mathbf{x})$ is defined by $F_i(\mathbf{x}) = max(F_{ij}(\mathbf{x}))$, where $1 \leq j \leq N_i$. The new point x is classified as follows: $G(\mathbf{x}) = \underset{i=1,\ldots,N}{argmax}(F_i(\mathbf{x}))$.

Theorem 3.1 (Equivalence)**.** *Given a new object* $\mathbf{x}$*, if one of the following conditions is true:*

(1) $\exists i, j (1 \leq i \leq N, 1 \leq j \leq N_i)$, $F_{ij}(\mathbf{x})_{Voting} = \sum_{k=1, k \neq i}^{N} N_k$;

(2) $\exists i, j (1 \leq i \leq N, 1 \leq j \leq N_i)$, $F_{ij}(\mathbf{x})_{MinMax} > 0$;

(3) $\exists i, j (1 \leq i \leq N, 1 \leq j \leq N_i)$, $F_{ij}(\mathbf{x})_{Prob} > F_{kp}(\mathbf{x})_{Prob}$, *where* $k \neq i$;

then $G(\mathbf{x})_{Voting} = G(\mathbf{x})_{MinMax} = G(\mathbf{x})_{Prob}$.

Proof: We first demonstrate the equivalence between these three sufficient conditions, and then prove that $G(\mathbf{x})_{Voting} = G(\mathbf{x})_{MinMax} = G(\mathbf{x})_{Prob}$ if the first condition is satisfied. **Equivalence**: Suppose the first condition is true: $F_{ij}(\mathbf{x})_{Voting} = \sum_{k=1, k \neq i}^{N} N_k$. It implies that the pseudo class C_{ij} wins all competitions against others. Because $\forall k, p (1 \leq k \leq N, k \neq i, 1 \leq p \leq N_k)$, $f_{ij-kp} > 0$, we have that $F_{ij}(\mathbf{x})_{MinMax} = min(f_{ij-kp}(\mathbf{x})) > 0$. This justifies the second condition. In addition, the fact $f_{ij-kp} > 0$ indicates that $F_{ij}(\mathbf{x})_{Prob} > F_{kp}(\mathbf{x})_{Prob}$. Thus, the third condition is also satisfied. The reverse can be proven in a similar manner.

Sufficiency: Suppose the first condition is true. It implies that $\forall k, p (1 \leq k \leq N, k \neq i, 1 \leq p \leq N_k)$, $f_{ij-kp}(\mathbf{x}) > 0$ and $f_{kp-ij}(\mathbf{x}) < 0$. Then, $F_{kp}(\mathbf{x})_{Voting} <= \sum_{o=1, o \neq k}^{N} N_o - 1$. Because $\frac{F_{ij}(\mathbf{x})_{Voting}}{\sum_{o=1, o \neq i}^{N} N_o} > \frac{F_{kp}(\mathbf{x})_{Voting}}{\sum_{o=1, o \neq k}^{N} N_o}$, we have that $F_{ij}(\mathbf{x})_{Voting} > F_{kp}(\mathbf{x})_{Voting}$, and $G(\mathbf{x})_{Voting} = i$. The condition $f_{ij-kp}(\mathbf{x}) > 0$ is identical to the condition $Prob(y = C_{ij}|\mathbf{x}) > Prob(y = C_{kp}|\mathbf{x})$, which implies that $F_{ij}(\mathbf{x})_{Prob} > F_{kp}(\mathbf{x})_{Prob}$ and $G(\mathbf{x})_{Prob} = i$. In addition, based on the preceding deviations, because $f_{ij-kp}(\mathbf{x}) > 0$, we have that $F_{ij}(\mathbf{x})_{MinMax} > 0$ and $F_{kp}(\mathbf{x})_{MinMax} < 0$. That means, except for the pseudo class C_{ij}, the decision functions of all other classes are smaller than zero, and $G(\mathbf{x})_{MinMax} = i$. Therefore, $G(\mathbf{x})_{Voting} = G(\mathbf{x})_{MinMax} = G(\mathbf{x})_{Prob}$.□

The inequivalence between these combination schemas may occur in the case of conflicts when the pre-condition of the above theorem is not satisfied. A conflict happens if inconsistent conclusions can be derived based on the pairwise decisions (or probabilities). For example, in Figure 1, suppose there are two classes C_1 and C_2. C_1 contains two subclasses $\{C_{11}, C_{12}\}$, and C_2 contains two subclasses $\{C_{21}, C_{22}\}$. Given a new object $\mathbf{x}$, suppose the pairwise probabilities have the relations: $Prob(y = C_{11}|\mathbf{x}) > Prob(y = C_{22}|\mathbf{x})$, $Prob(y = C_{22}|\mathbf{x}) > Prob(y = C_{12}|\mathbf{x})$, $Prob(y = C_{12}|\mathbf{x}) > Prob(y = C_{21}|\mathbf{x})$, and $Prob(y = C_{21}|\mathbf{x}) > Prob(y = C_{11}|\mathbf{x})$. From the last three relations, we have that $Prob(y = C_{22}|\mathbf{x}) > Prob(y = C_{11}|\mathbf{x})$, which is in conflict with the first relation $Prob(y = C_{11}|\mathbf{x}) > Prob(y = C_{22}|\mathbf{x})$. To handle conflicts, the voting-based schema utilizes the strategy of majority vote to determine the subclass with the maximum posterior probability; the probability-based schema directly estimates the posterior probabilities by using the Kullback-Leibler (KL) distance as the loss function [11]; the MinMax schema selects one representative prototype classifier for each subclass and then uses their decision values to determine the maximum probable subclass to be returned.

These three schemas, as well as their equivalence, are illustrated in Figure 1. There are two classes $\{C_1, C_2\}$, and their subclasses are $\{C_{11}, C_{12}\}$ and $\{C_{21}, C_{22}\}$, respectively. For each object $\mathbf{x}$ inside the region *ABCNMA*, the sub-

class C_{11} wins the competitions against the subclasses C_{21} and C_{22}. Then, $F_{11}(\mathbf{x})_{Voting} = 2$, and $G(\mathbf{x})_{Voting} = C_1$. Because $f_{11-22}(\mathbf{x}) > 0$ and $f_{11-21}(\mathbf{x}) > 0$, $F_{11}(\mathbf{x})_{MinMax} > 0$ and $G(\mathbf{x})_{MinMax} = C_1$. Also, because $Prob(y = C_{11}|\mathbf{x})$ is larger than $Prob(y = C_{21}|\mathbf{x})$ and $Prob(y = C_{22}|\mathbf{x})$, $G(\mathbf{x})_{Prob} = 2$. Therefore, the three schemas are equivalent inside the region *ABCNMA*. Similarly, the equivalence is held in the regions *CDEONC*, *EFGPOE*, and *GHAMPG*. However, inside the small center region *MNOPM*, the above conditions are not satisfied and therefore the equivalence is not guaranteed.

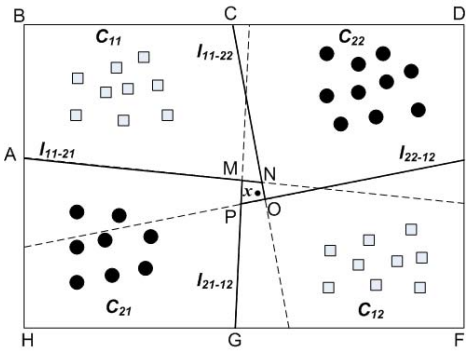


Figure 1: An example of combination schemas

Theorem 3.1 indicates that the three combination schemas are equivalent inside certain regions. As shown in Section 5.1, we empirically verified that these equivalent regions occupy in overall more than 99% of the whole input space. That means, these combination schemas are equivalent in most cases. It explains why different combination schemas usually exhibit similar accuracy.

Another observation is that, since the conflicts rarely happen in practice, we can reasonably assume that Theorem 3.1 is true for the whole space. Under this assumption, the pairwise coupling becomes equivalent to a voronoi diagram problem [18]. Particularly, each subclass (C_{ij}) has a dominated region (voronoi polytope), which is bounded by a subset of the related linear prototype classifiers (separating hyperplanes). If a new object $\mathbf{x}$ is within the dominated region of C_{ij}, then it is classified to the class C_i. Thus, the pairwise coupling problem can be re-formulated as: "*Given a new object* $\mathbf{x}$*, search for a class region (voronoi polytope), which contains the object* $\mathbf{x}$." Based on this reformulation, traditional voronoi techniques [18] can be conveniently adapted to identify the dominated region for each subclass. The significant (necessary) and insignificant (redundant) prototype classifiers can also be identified. Redundant prototype classifiers refer to the prototype classifiers that do not contribute to the decision boundary of the resulting combined classifier. In addition, spatial indexing structures (e.g., R-tree) can be utilized to index the subclass regions, such that the classification time cost can be significant reduced.

3.2 The Impact of Cluster Granularity

Cluster granularity is an important parameter for most existing clustering algorithms and is usually predefined by users. For the problem of LLC-PC, there is a tradeoff between cluster granularity and generalization accuracy. If the cluster granularity increases, more local features of the training set will be considered in the resulting classifier. Conversely, if the cluster granularity decreases, more global features will be considered. As a result, a low granularity of clusters may lead to the problem of underfitting, whereas a high granularity may contribute to the problem of overfitting. It is important to delve into their relationship, which would provide a theoretical foundation to guide the possible solutions of LLC-PC.

We first study two extreme cases. At the lowest clustering granularity, each class has only one single cluster. In this case, the resulting classifier is equivalent to a linear classifier. Because the training set is nonlinearly separable, it is impossible to find a linear decision boundary that can correctly classify all the sample objects. The data set will be underfitted. In another extreme case, where the clusters have the highest granularity, each sample object is regarded as a single cluster. As proved by Theorem 3.2, the resulting classifier is equivalent to a 1-nearest neighbor classifier, which classifies a new object $\mathbf{x}$ based on its closest sample object in the training set. According to the well-known characteristics of 1-nearest neighbor classifiers [12], the resulting classifier can easily lead to overfitting.

Theorem 3.2. *Given a data set of $N(N > 1)$ classes, if the maximum number of subclasses are generated for each class, then the resulting classifier is equivalent to a 1-nearest-neighbor classifier.*

Proof. First, consider the case when the voting-based is selected as the combination schema. At the highest cluster granularity, each class C_{ij} corresponds to a single object $\mathbf{x}_{ij}$. Given a new object $\mathbf{x}$, suppose its nearest object is $\mathbf{x}_{ij}$. Clearly, a 1-nearest-neighbor classifier will classify $\mathbf{x}$ to the class C_i. Because the optimal separating hyperplane for classes C_{ij} and C_{kp} is the bisection hyperplane between objects $\mathbf{x}_{ij}$ and $\mathbf{x}_{kp}$, we have that $\forall k, p(1 \leq k \leq N, k \neq i, 1 \leq p \leq N_k)$, $\mathbf{x}$ is closer to $\mathbf{x}_{ij}$ than to $\mathbf{x}_{kp}$. That means $f_{ij-kp}(\mathbf{x}) > 0$ and $F_{ij}(\mathbf{x}) = \sum_{o=1,o\neq i}^{N} N_o$. The condition $f_{kp-ij}(\mathbf{x}) < 0$ implies that $F_{kp} < \sum_{o=1,o\neq k}^{N} N_o$, since there exists at least one sample object $\mathbf{x}_{ij}$ that does not belong to the class C_k and $\delta(f_{ij-kp}(\mathbf{x})) = 0$. Therefore, $F_k(\mathbf{x})_{Voting} < F_i(\mathbf{x})_{Voting} = 1$. $\mathbf{x}$ is classified as the class C_i, the same result of the 1-nearest neighbor classifier. Second, consider the case when the probability-based or MinMax is used as the combination schema. As derived in the preceding, $F_{ij}(\mathbf{x}) = \sum_{o=1,o\neq i}^{N} N_o$. From theorem 3.1, in this case, the three combinations schemas are equivalent.□

We have conducted extensive experiments on twenty benchmark data sets. Figure 2 shows two abstract patterns found based on the experimental results, which are consistent with the preceding theoretical evaluations. Pattern a) shows a relationship similar to the well-known Bias-Variance Dilemma. When the number of clusters increases, the prediction accuracy will first increase to a certain level, then continuously decrease, and finally stabilize at a certain level. That means for a small number of clusters, this approach could increase the accuracy of a linear classifier. However, arbitrarily increasing the number of clusters would have a nontrivial impact on the classification accuracy. Pattern b) shows a different relationship. There exist some data sets, where generating subclasses will only deteriorate the classifier performance. When the number of clusters increases, the classification accuracy will first decrease rapidly, then decline more slowly, and finally stablize at a certain level.

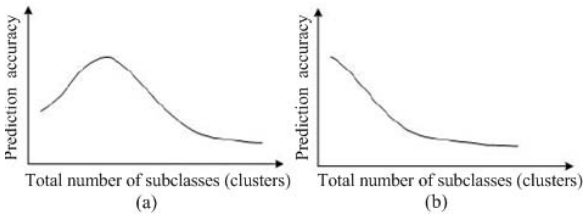


Figure 2: Two patterns related to cluster granularity

4 Criterion Functions

This section addresses the criterion functions which can measure the generalization accuracy of the combined classifier, by considering a number of factors, such as the division of original classes, the binary classifier model, the combination schema, and the computational cost. Many existing methods directly use general clustering criterion functions (e.g., total intra-cluster variance [17]) to measure the quality of the subclasses generated. However, the subclasses that minimize the total intra-cluster variance do not necessarily lead to the classifier of a high generalization accuracy.

For example, Figure 3 shows a data set of two classes (circle and square) and the dotted curve refers to the true decision boundary. For simplicity, we only conducted the clustering process on the circle class. Figures 3 (a) and 3 (b) show the results of two different partition strategies. The estimated decision boundary led by the right partition is much more accurate than that by the left partition, even its total intra-cluster variance is much larger than the left partition. From the view point of classification, a good criterion function should be able to measure both the accuracy attained on the training set and the structure capacity, that is, the ability of the classifier to correctly predict class labels for future instances. Following this direction, several new criterion functions are presented.

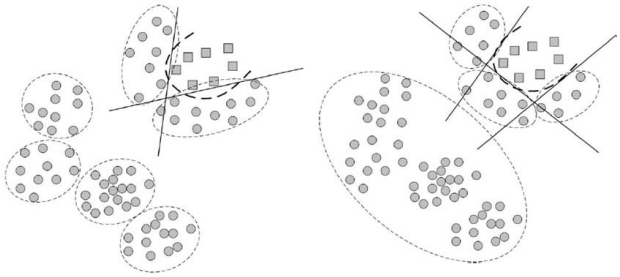


Figure 3: An example of two different partitions

4.1 Mean Piecewise Error Function

The mean piecewise error function can be formalized as:

$$Q = \sum_{(C_{ij},C_{kp})\in \mathbf{U}} \Big(P_{ij-kp} E(C_{ij},C_{kp}) \Big), \tag{1}$$

where $\mathbf{U} = \{(C_{ij},C_{kp}) | 1 \le i,k \le N, i \ne k, 1 \le j \le N_i, 1 \le p \le N_k\}$, N refers to the total number of original (parent) classes, N_i refers to the number of subclasses for the parent class C_i, P_{ij-kp} denotes the prior probability of the subclass pair (C_{ij},C_{kp}), and $E(C_{ij},C_{kp})$ denotes the generalization error between subclasses C_{ij} and C_{kp}. The prior probabilities are used as the weights to balance the contributions of different subclasses. We set $P_{ij-kp} = \frac{P_{ij}\cdot P_{kp}}{\sum_{(C_{ij},C_{kp})\in \mathbf{U}} P_{ij}\cdot P_{kp}}$, where $P_{ij} = |C_{ij}|/S$, the ratio of the sample size of subclass C_{ij} to the total sample size.

The selection of the atomic error function $E(C_{ij},C_{kp})$ depends on the binary classifier model used for the subclasses C_{ij} and C_{kp}. We consider two popular linear classifier models, including Fisher linear discriminant analysis (LDA) and linear support vector machines (SVM). We select an identical classifier model for each pair of subclasses with default parameter settings. Depending on the specific classifier model selected, we abbreviate the related mean piecewise error (MPE) function as MPE-SVM or MPE-LDA. The whole category of MPE functions is abbreviated as MPE.

MPE-LDA selects the inverse of Fisher criterion [17], the ratio of the between-class variance to the within-class variance, as the atomic error function. It can be formalized as follows:

$$Q = \sum (P_{ij-kp} (\mathbf{w}_{ij-kp}^{\mathrm{t}} \mathbf{S}_{W,ij-kp} \mathbf{w}_{ij-kp}) (\mathbf{w}_{ij-kp}^{\mathrm{t}} \mathbf{S}_{B,ij-kp} \mathbf{w}_{ij-kp})^{-1}) \tag{2}$$

, where $\mathbf{S}_{W,ij-kp} = \mathbf{S}_{ij} + \mathbf{S}_{kp}$ and $\mathbf{S}_{B,ij-kp} = (\mathbf{m}_{ij} - \bar{\mathbf{m}})(\mathbf{m}_{ij} - \bar{\mathbf{m}})^{\mathrm{t}} + (\mathbf{m}_{kp} - \bar{\mathbf{m}})(\mathbf{m}_{kp} - \bar{\mathbf{m}})^{\mathrm{t}}$ are the within-class scatter matrix and the between-class scatter matrix, respectively; $\mathbf{S}_{ij}$ is the within-class covariance matrix of subclass C_{ij}, $\mathbf{m}_{ij}$ is the mean vector of subclass C_{ij}, similar definitions are used for $\mathbf{S}_{kp}$ and $\mathbf{m}_{kp}$, $\bar{\mathbf{m}} = (\mathbf{m}_{ij} + \mathbf{m}_{kp})/2$, and $\mathbf{w}_{ij-kp} = \mathbf{S}_{W,ij-kp}^{-1}(\mathbf{m}_{ij} - \mathbf{m}_{kp})$. The definitions of other symbols are consistent with the related definitions for Equation (9).

MPE-SVM selects the error function of a linear SVM model, the addition of the inverse classifier margin to the empirical error, as the atomic error function. It can be formalized as follows:

$$Q = \sum\left(P_{ij-kp}\frac{1}{2}\|\mathbf{w}_{ij-kp}\|^2\right) + C\sum\left(P_{ij-kp}\sum_{o=1}^{m_{ij-kp}}\zeta_{o,ij-kp}\right), \quad (3)$$

where $\frac{1}{2}\|\mathbf{w}_{ij-kp}\|^2$ and $\zeta_{o,ij-kp}$ refer to the inverse classifier margin and the slack variables for subclasses C_{ij} and C_{kp}, respectively; m_{ij-kp} refers to the number of slack variables, and C denotes the tradeoff parameter. For simplicity, we assume that the tradeoffs of all SVM classifiers are identical. The left part of the equation is the weighted sum of the inverse margins of pairwise SVM classifiers, which can be regarded as the approximate structure error of the combined classifier. The right part of the equation is the weighted sum of the slack variables of pairwise SVM classifiers, which can be viewed as the approximate empirical error of the combined classifier. The parameter C is used to balance the contributions of the classifier margin and the empirical error.

4.2 Major Characteristics

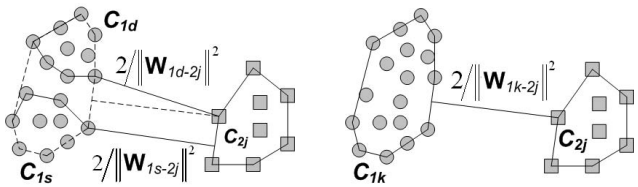
This subsection first evaluates the correlation between the proposed criterion functions and the cluster granularity, and then conducts a comparison between these criterion functions.

Theorem 4.1 (Monotonicity of MPE-SVM)**.** *Given a data set of N classes $(C_1,...,C_N)$, suppose each class C_i has N_i subclasses, then the value of MPE-SVM can be decreased by randomly decomposing one subclass into two smaller-size subclasses.*

Proof Sketch: For simplicity, we only consider the case of binary classes (N = 2), and prove the theorem in two different scenarios. The case N $>$ 2 can be proved similarly.

Linearly Separable Case. Suppose each pair of subclasses C_{1i} and C_{2j} are linearly separable. Then there is no classification error for each piecewise SVM and only margins need to be considered. The theorem becomes: "The average margin will always increase if randomly decompose a cluster into two clusters." We prove it by using the concept of convex hull [15]. As shown in Figure 4, suppose the subclass C_{1k}, $1 \le k \le N_1$, is randomly selected and partitioned into two subclasses C_{1d} and C_{1s}. Their corresponding prior probabilities are p_{1d} and p_{1s}, respectively. $p_{1d} + p_{1s} = p_{1k}$. For any subclass C_{2j} of the class C_2, according to the characteristics of convex hull , $\frac{1}{2}\|w_{1d-2j}\|^2 < \frac{1}{2}\|w_{1k-2j}\|^2$, and $\frac{1}{2}\|w_{1s-2j}\|^2 < \frac{1}{2}\|w_{1k-2j}\|^2$. Then $p_{1d}\cdot p_{2j}\cdot \frac{1}{2}\|w_{1d-2j}\|^2 + p_{1s}\cdot p_{2j}\cdot\frac{1}{2}\|w_{1s-2j}\|^2 < p_{1k}\cdot p_{2j}\cdot\frac{1}{2}\|w_{1k-2j}\|^2$. It can be deduced that $\sum_j(p_{1d}\cdot p_{2j}\cdot\frac{1}{2}\|w_{1d-2j}\|^2 + p_{1s}\cdot p_{2j}\cdot\frac{1}{2}\|w_{1s-2j}\|^2) + \sum_{i,j,i\neq k}(p_{1i}\cdot p_{2j}\cdot\frac{1}{2}\|w_{1i-2j}\|^2)) < \sum_j(p_{1k}\cdot p_{2j}\cdot\frac{1}{2}\|w_{1k-2j}\|^2 + \sum_{i,j,i\neq k}(p_{1i}\cdot p_{2j}\cdot\frac{1}{2}\|w_{1i-2j}\|^2))$.

Linearly Nonseparable Case. Suppose at least one pair of subclasses C_{1i} and C_{2j} are nonlinearly separable. Then it is necessary to consider both margin and slack variables. Let $M(C_{1i},C_{2j})$ and $S(C_{1i},C_{2j})$ denote the inverse margin and the set of slack variables between C_{1i} and C_{2j}, respectively. Let $Q(C_{1i},C_{2j}) = M(C_{1i},C_{2j}) + Sum(S(C_{1i},C_{2j}))$. A subclass C_{1k} is randomly selected from from C_1, $1 \le k \le N_1$, and partitioned it into two clusters C_{1d} and C_{1s}. Given any subclass C_{2j} of the class C_2, if the same separating hyperplane and margin are used for C_{1k} and C_{2j} as the candidate hyperplane (M^*) and margin (S^*) for the classes C_{1d} and C_{2j}, then $S(C_{1k},C_{2j}) = S^*(C_{1d},C_{2j})\bigcup S^*(C_{1s},C_{2j})$. $p_{1k}\cdot p_{2j}\cdot Q(C_{1k},C_{2j}) \ge p_{1d}\cdot p_{2j}\cdot Q^*(C_{1d},C_{2j}) + p_{1s}\cdot p_{2j}\cdot Q^*(C_{1s},C_{2j})$. The optimal separating hyperplane for classes C_{1d} and C_{2j} can achieve equal or smaller classification error than $Q^*(C_{1d},C_{2j})$. That is: $Q(C_{1d},C_{2j}) \le Q^*(C_{1d},C_{2j})$. Similarly, $Q(C_{1s},C_{2j}) \le Q^*(C_{1s},C_{2j})$. Then, $p_{1d}\cdot p_{2j}\cdot Q(C_{1d},C_{2j}) + p_{1s}\cdot p_{2j}\cdot Q(C_{1s},C_{2j}) \le p_{1k}\cdot p_{2j}\cdot Q(C_{1k},C_{2j})$. We have that $\sum_j(p_{1d}\cdot p_{2j}\cdot Q(C_{1d},C_{2j}) + p_{1s}\cdot p_{2j}\cdot Q(C_{1s},C_{2j})) + \sum_{i,j,i\neq k}(p_{1i}\cdot p_{2j}\cdot Q(C_{1i},C_{2j})) \le \sum_j(p_{1k}\cdot p_{2j}\cdot Q(C_{1k},C_{2j})) + \sum_{i,j,i\neq k}(p_{1i}\cdot p_{2j}\cdot Q(C_{1i},C_{2j}))$. □



(a) Before partition (b) After partition

Figure 4: An example of Theorem 4.1

Theorem 4.2. *Given a data set of N classes, the criterion functions MPE-SVM and MPE-LDA are minimized if the maximum number of subclasses are generated for each class.*

Proof. According to Theorem 4.1, given k subclasses that have been generated by any clustering algorithm, we can always find $k+1$ subclasses that can achieve smaller MPE-SVM than the k subclasses. By recursive theorem, the score MPE-SVM can be minimized at the highest cluster granularity. Consider the case of MPE-LDA, because at the highest cluster granularity, each training object is regarded as a subclass and the variance of a single object equals zero, the score MPE-LDA is identical to zero (the minimal value).□

MPE-LDA vs. MPE-SVM

First, we consider the case when each class only contains one cluster (the lowest cluster granularity). In this case, these two functions degeneralize to LDA and SVM, respectively. Results have been shown that in overall SVM can achieve higher classification accuracy than LDA [17]. The possible reason is that SVM considers both empirical error and structure capacity and is based on recent advances

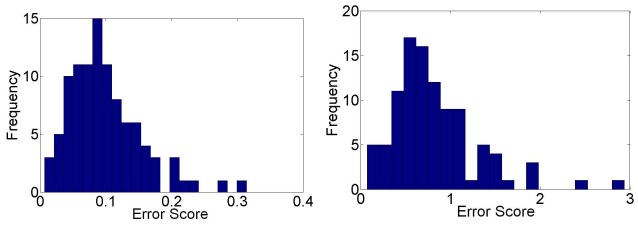
in statistical learning theory [16]. In comparison, LDA assumes that each class is normally distributed with common covariances. This assumption is usually not held in real applications. However, LDA is much more efficient to compute and easier to understand than SVM. Particularly, LDA and SVM have the time complexities $O(d^2n)$ and $O(d^2n^\delta)$, respectively, where d refers to the dimension cardinality, n refers to the training sample size, and $\delta > 1$.

Second, we consider the case when some classes have more than one subclass. In this case, MPE-SVM appears more stable than MPE-LDA. As proven by Theorem 4.1, MPE-SVM has the important characteristic of monotonicity, with respect to the total number of subclasses. It is also more resilient to outliers. In comparison, MPE-LDA does not have the feature of monotonicity (See examples in Section 6.4) and requires calculating the inverse of the within-class scatter matrix for each pair of subclasses. If some subclasses have singular covariance matrixes (e.g., outlier classes or the classes with correlated attributes), then the total score of MPE-LDA will be affected. The selection of MPE-LDA or MPE-SVM is not necessarily dependent on the classifier model used in the pairwise prototype classifiers. For example, in the scenario of limited computation, MPE-LDA may be used as the criterion function to guide the generation of subclasses, even though SVM is used latter to build the pairwise prototype classifiers.

Characteristics of MPE

As demonstrated in Section 6, MPE exhibits much higher accuracy than general clustering criterion functions (e.g., total intra-cluster variance). However, it still has several limitations: 1) **Dependence on Cluster Granularity**. Theorem 4.2 indicates that MPE can always get the minimal value at the highest cluster granularity. As proved by Theorem 3.2, in this case the combined classifier degeneralizes to a 1-nearest-neighbor classifier. It implies the requirement of a predefined total number of subclasses to be generated. Otherwise, the criterion functions may not be useful to find meaningful subclasses.

2) **Inappropriate for a Large Number of Subclasses**. The total number of prototype classifiers is quadratically increasing with the increase of the total number of subclasses. When the number is high, the differences between the error scores of prototype classifiers will be neutralized. As a result, MPE will become insensitive to different generalizations of subclasses. As shown in Figure 5, this effect is demonstrated on a binary class benchmark data ringnorm, which has twenty dimensions and 400 training instances. We randomly generated 20 subclasses (10 subclasses for each class). The histograms of the error scores (calculated by LDA and SVM, respectively) of pairwise prototype classifiers are displayed in Figures 5 (a) and 5 (b). K refers to the total number of subclasses generated. The results show that the pairwise error scores are not well-differentiable. We further ran 100 different random generalizations of subclasses and obtained similar distributions in most cases. It indicates that the value of MPE will not change significantly given different generalizations of subclasses.



(a) LDA as the classifier model (b) SVM as the classifier model

Figure 5: Histograms of the error scores of pairwise prototype classifiers (Ringnorm, K = 20)

4.3 Variants of MPE

To alleviate the negative impacts of the large number of prototype classifiers, we can redefine the set **U** (see equation (1)) as a small set of representative prototype classifiers. Depending on the different definitions of the representative classifiers, several variants of MPE can be derived. Due to lack of space, we only briefly present two major variants.

The first variant is called **R**efined MPE (R-MPE), which defines **U** as the set of necessary prototype classifiers. As discussed in Section 3.1, by assuming that Theorem 3.1 is true for the whole space, the pairwise coupling can be reformulated as a voronoi diagram problem. Based on this reformulation, many prototype classifiers are actually redundant when the data is in a low-dimensional space (e.g., smaller than 10 dimensions). For example, suppose there are totally N subclasses in a 2-dimensional space, then the number of necessary prototype classifiers is smaller than $(3N-6)$ [18]. That means, even there are $O(N^2)$ prototype classifiers, only linearly many classifiers contribute to the decision boundary of the resulting classifier.

Another variant is named **S**ymmetric **N**earest **N**eighbor based MPE (SNN-MPE), which defines **U** as the pairs of subclasses which are symmetric k-nearest neighbors. We define the Euclidean distance between the centers of two subclasses as the proximity metric. The subclasses of a same parent class are not considered as neighbors. The effectiveness of SNN-MPE is based on an important observation that the significant prototype classifiers are usually related to the pairs of subclasses, which are close to each other. SNN-MPE provides a parameter k to allow users to balance the tradeoff between the computational cost and the accuracy.

The proposed variants R-MPE and SNN-MPE are both mainly appropriate for low-dimensional data. Because of the curse of dimensionality, in a high dimensional space the subclasses are neighbors to each other, and most pairwise prototype classifiers become necessary classifiers. In this case, R-MPE degeneralizes to MPE, and SNN-MPE degeneralizes to a random selection of subclass pairs for **U**. In

order to apply these criterion functions to high-dimensional data, we can utilize dimension reduction techniques (e.g., PCA, LDA) to reduce the dimensionality.

5 A Greedy Clustering Algorithm

To evaluate the effectiveness of the proposed criterion functions, this section presents a simple but effective supervised clustering algorithm named Greedy-MPE. It generates the subclasses in a greedy manner to minimize the criterion functions (MPE). The algorithm is described as follows:

Algorithm (Greedy-MPE). Given a data set of N classes $\{C_1,\ldots,C_N\}$ and the total number (K) of subclasses to be generated,

1. Regard each class as a single cluster (subclass).
2. From the set **U** of subclass pairs, search for a pair of subclasses (C_{ij},C_{kp}) that has the maximum weighted classification error $F(C_{ij},C_{kp})$. The maximum weighted classification error indicates that this pair of subclasses is currently most linearly inseparable and hence can be regarded as the priority candidate subclasses for further decompositions.
3. Select a subclass from C_{ij} and C_{kp}, which has the highest intra-class variance, and decompose it into two smaller-size subclasses.
4. If the total number of the subclasses generated is smaller than K, then go to step 2. Otherwise, output the current subclasses and terminate the algorithm.

The set **U** of candidate subclass pairs is determined by a specific criterion function, which the algorithm greedily minimizes. For example, for MPE, **U** refers to the pairs of subclasses, which do not have the same parent class label. For SNN-MPE, **U** refers to the pairs of subclasses, which are symmetric k-nearest neighbors. $F(C_{ij},C_{kp}) = P_{ij-kp} * E(C_{ij},C_{kp})$, where P_{ij-kp} refers to the prior probability of the subclass pair (C_{ij},C_{kp}), and $E(C_{ij},C_{kp})$ refers to the classification error between C_{ij} and C_{kp}. In the step 3, traditional clustering algorithms (e.g., k-means) can be used to decompose the selected subclass into two smaller-size subclasses.

The key issue of Greedy-MPE is to select an appropriate subclass in each iteration for further splits. The current selection bias is to prefer the subclass which is not well-separatable from others and has a high intra-cluster variance. Two alternative selection biases may also be considered. The first is to prefer the subclass which has the highest aggregated classification error over the related subclass pairs: $\arg\max_{C_{ij}}(\sum_{k\neq i} P_{ij-kp}E(C_{ij},C_{kp}))$. The second is to prefer the subclass which has the maximum gain of MPE score: $\arg\max_{C_{ij}}(Q_{before\_splitting\_C_{ij}} - Q_{after\_splitting\_C_{ij}})$, where $Q_{before\_splitting\_C_{ij}}$ refers to the MPE score before splitting the subclass C_{ij}, and $Q_{after\_splitting\_C_{ij}}$ refers to the MPE score after splitting the subclass C_{ij}.

6 Experiment

This section demonstrates the equivalence between three popular combination schemas under general settings (Theorem 3.1), compares the performances of the resulting classifiers produced by different clustering methods, and evaluates the application of the proposed criterion function MPE-SVM to estimate the appropriate number of subclasses for different generation methods.

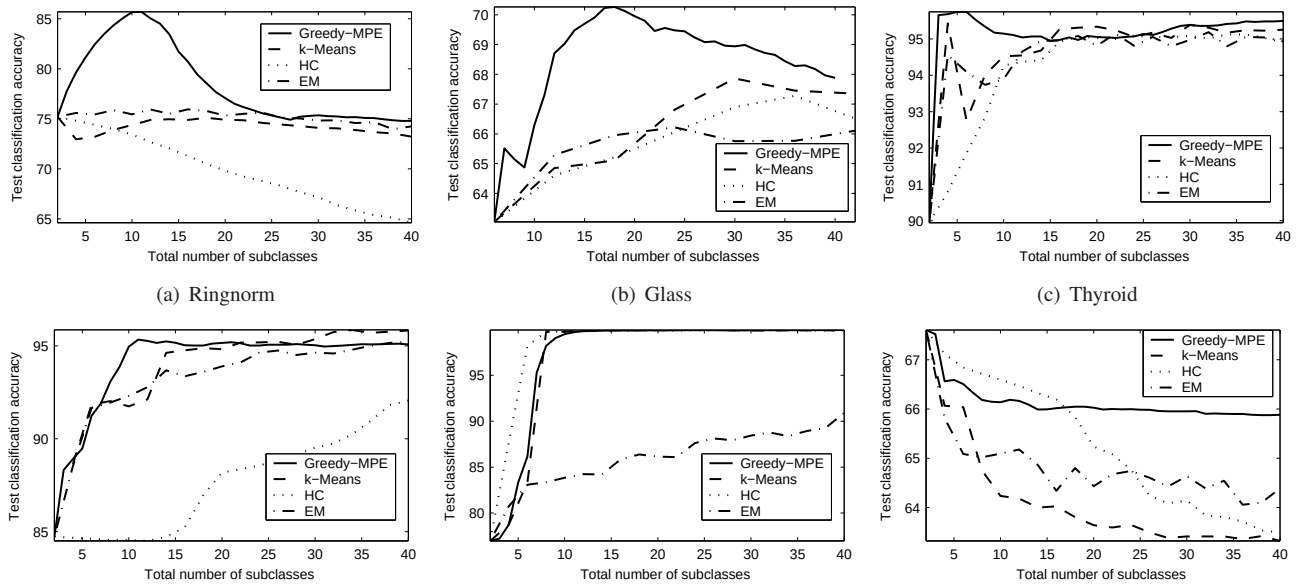
6.1 The Experimental Setup

Experimental Tools. We used linear SVM and Fisher LDA as the prototype classifiers, and four different clustering algorithms to generate subclasses: Greedy-MPE, k-means, hierarchical clustering (HC), and EM clustering. The implementation of linear SVM is LIBSVM [20]. The clustering algorithms k-means, HC and EM were implemented in Matlab. The major settings were as follows: 1) Euclidean distance was used as the proximity metric, 2) the parameter "replicates" for k-means (number of times to repeat the clustering) was set to 10, 3) the link metric in the HC clustering algorithm was set to average link, and 4) the tradeoff parameter (C) for linear SVM was set to 100. The default classifier model and combination schema were linear SVM and the voting-based, respectively. For k-means, HC and EM, we generated the same number of subclasses for each class.

Experimental Data Sets. In our experiments, we used 22 benchmark data sets provided by UCI [19], STATLOG [21], DELVE [22], and LIBSVM [20] data repositories: flare solar, thyroid, breast cancer, breast-w, pima-diabetes, heart, image, ringnorm, twonorm, waveform, german, diabetis, fourclass, svmguide1, vehicle, page-block, segment, glass, satimage, pendigits, optdigits, and letter. Among these data sets, the range of class numbers is [2, 26], and the range of dimensions is [2, 60]. Table 1 shows the detailed information of six representative data sets. We generated 100 random partitions into training and test sets (mostly 60%:40%). On each partition, we trained a classifier and then calculated its test set accuracy. The mean accuracy over all partitions was reported. We considered the settings of cluster granularity (the total number of subclasses) from 1 to 40. The experiments were conducted on the Matlab 6.5 framework running on Windows XP. The hardware platform was a 2.8 GHz Pentium-D CPU with 1GB of RAM.

6.2 Combination Schemas

This subsection validates the equivalence between three popular combination schemas (voting based, probability based, and MinMax) on the generalization accuracy. As discussed in Section 3, these three combinations are provably equivalent inside certain regions, that constitute a majority of the input space. To evaluate the percentage of the provable equivalent area to the whole space, we used k-means to



(a) Ringnorm (b) Glass (c) Thyroid

(d) Image (e) Fourclass (f) Flare solar

Figure 7: Comparison on test classification accuracy

Table 1: Some characteristics of experimental data sets

Dataset	Source	#Objects	#features	#classes
Thyroid	UCI	140:75	3	2
Flare solar	UCI	666:400	9	2
Image	UCI	1300:1010	18	2
Glass	UCI	128:86	9	6
Ringnorm	DELVE	400:7000	20	2
Fourclass	LIBSVM	517:345	2	2

Note: The numbers before and after ":" are for training and testing, respectively.

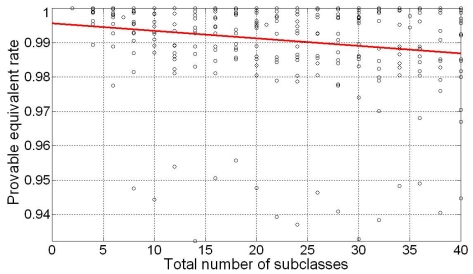


Figure 6: Provable equivalent rate vs. cluster granularity

generate subclasses and calculated the rate of training and testing objects, which were within the provable equivalent area. Figure 6 shows the experimental results on the twenty-two benchmark data sets. The *X-axis* refers to the total number of subclasses generated and the *Y-axis* refers to the rate of training and testing objects which are within the provable equivalent area. In the figure, there are totally 306 sample points, and each sample point denotes the result of a data set under a specific cluster granularity. A linear regression line was generated to show the correlation between the provable equivalent rate and the cluster granularity. The results indicate that on average more than 99% of objects are within the provable equivalent area. Another observation is that the provable equivalent rate has a tendency of decreasing when the cluster granularity increases. That means, when the cluster granularity is extremely high (e.g., 200), these schemas will be significantly different. However, as shown later, the optimal number of subclasses is usually smaller than 40 in practice.

Theorem 3.1 is the sufficient but not necessary condition of the equivalence. The objects which do not satisfy Theorem 3.1 are still possibly equivalent for these combination schemas. We observe that the actual equivalent rate is much higher than the provable equivalent rate. For example, among all the tested data sets, the actual equivalent rate between the voting based and MinMax is 0.999 ± 0.002. As to the non-equivalent objects, in which the voting based and MinMax reported different results, these two schemas have the test accuracies close to a random assignment. For instance, among fourteen binary data sets, the voting based and MinMax schemas have the test accuracies of 0.52 ± 0.33 and 0.48 ± 0.33, respectively, on the non-equivalent objects.

6.3 Subclass Generation

This subsection compares the performances of the LLC-PC classifiers led by Greedy-MPE and three popular clustering algorithms, k-means, HC, and EM. Figure 7 shows the results on test classification accuracy. The *X-axis* refers to the total number of subclasses, and the *Y-axis* refers to the test accuracy. Table 2 shows the comparison on optimal test accuracy, which refers to the highest test accuracy

over all the settings. The results indicate that Greedy-MPE is more accurate and stable than general clustering algorithms in most settings. For example, on the data set ringnorm, the optimal test accuracy of Greedy-MPE is 10% higher than those of the other algorithms. A possible explanation to this superiority is that Greedy-MPE is guided by the criterion function MPE. Because MPE is specifically designed to measure the generalization error of an LLC-PC classifier, a greedy division of the training data to minimize MPE can be regarded as a greedy division strategy to minimize the generalization error. Thus, the overall good (but not optimal) accuracy and stability are guaranteed.

In comparison, general clustering algorithms exhibit inconsistent performances on different data sets. For example, the EM clustering algorithm can achieve comparable optimal test accuracies to the others on image and thyroid, however, its performances on ringnorm and fourclass are 10% less than Greedy-MPE. As shown in Figure 7, this pattern of inconsistency is also exhibited in other settings. It is important to compare the algorithms over all the settings, since in practice it is difficult to accurately estimate the optimal number of subclasses. A possible explanation of this inconsistency is that the criterion functions (e.g., total intra-cluster variance) of general clustering algorithms are not well-correlated to the generalization accuracy. As a result, the generated subclasses with high cluster quality do not necessarily mean that the resulting classifier will have a high generalization accuracy.

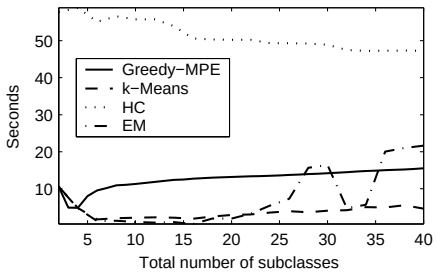


Figure 8: Comparison on total computational cost (Image)

Table 2: Comparison on optimal test accuracy

Dataset	Greedy-MPE	k-Means	HC	EM
Ringnorm	85.70(11)	75.20(2)	75.20(2)	75.95(16)
Glass	70.27(18)	67.86(30)	67.28(36)	66.22(24)
Fourclass	99.94(38)	99.95(22)	99.93(26)	90.93(40)
Image	95.34(11)	95.86(34)	92.06(40)	95.22(40)
Thyroid	95.75(5)	95.43(4)	95.12(34)	95.19(32)
Flare solar	67.61(2)	67.61(2)	67.61(2)	67.61 (2)

Note: # in "()" refers to the optimal number of subclasses.

Figure 8 shows the time comparison results on the data set image. The *X-axis* refers to the total number of subclasses, subclasses, and the *Y-axis* refers to the total computational cost, which contains the clustering, training, and testing time costs. The total computational cost of the LLC-PC classifier led by Greedy-MPE is competitive to those of the classifiers led by k-means and EM, but much lower than that of the LLC-PC classifier generated by HC. Other tested benchmark data exhibit similar trends.

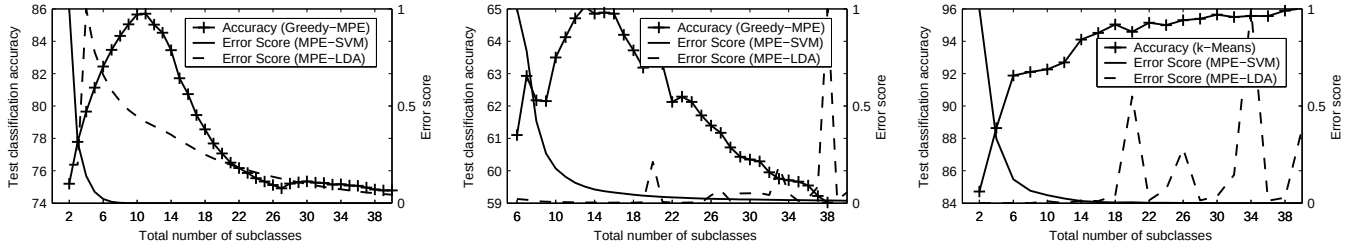
6.4 Estimating the Optimal Number of Subclasses

This subsection demonstrates the application of MPE-SVM to estimate the optimal number of subclasses for different subclasses generation methods. As illustrated in Figure 9 (a), we applied the LLC-PC classifier led by Greedy-MPE to the data set ringnorm. The left *Y-axis* refers to the test accuracy, the right *Y-axis* refers to the error score (calculated by MPE-SVM), and the *X-axis* refers to the total number of sub-classes. An important observation is that, if the total number of sub-classes increases, the error score will decrease rapidly in the beginning. After the error score reduces to a certain level, the decreasing speed will slow down and stabilize. Consistently, the change of the test accuracy matches this pattern. When the decreasing of the error score clearly slows down, the test accuracy approaches a close-to-optimal value. Therefore, we can estimate an appropriate number of sub-classes around 8, which is close to the optimal value 11.

This pattern was observed on all the data sets tested when different subclasses generation methods (k-means, HC) and classifier models (e.g., SVM, LDA) were used. For example, in Figure 9 (b), we used LDA as the classifier model, instead of SVM, and selected a different data set glass. The correlation between the error score MPE-SVM and the test accuracy indicates an appropriate number of subclasses around 14, which is close to the optimal value 13. In Figure 9 (c), we applied k-means, instead of Greedy-MPE, to the data set image, and could estimate an appropriate number of subclasses around 18. A possible explanation to this pattern is that MPE-SVM considers both classifier margin and empirical error and is hence well correlated to the generalization error. A rapid reduction on the MPE-SVM score would expect a significant decreasing on the generalization error. For the purpose of comparing MPE-SVM and MPE-LDA, we also showed the related error scores calculated by MPE-LDA in the preceding examples. By comparing the curves of MPE-SVM with those of MPE-LDA, the indicated patterns are consistent with the theoretical evaluations in Section 4.2.

7 Conclusion and Future Work

This paper conducts a systematic evaluation of LLC-PC and presents several criterion functions to measure the goodness of subclasses with respect to the generalization accuracy. We evaluate their major characteristics, and demonstrate how to apply these criterion functions to identify



(a) Ringnorm (b) Glass (LDA as the classifier model) (c) Image

Figure 9: Correlation between test classification accuracy and MPE

good-quality subclasses and to estimate the optimal number of subclasses. Extensive experimental evaluations further validated the effectiveness of the proposed techniques. In the future, we plan to conduct empirical comparisons between LLC-PC and other categories, LLC-LS and LLC-MD, and summarize the appropriate applications for each one. We will also study the theoretical connections between different categories and design a general framework for them.

References

[1] T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. In *Journal of the Royal Statistical Society*, (Series B), 58:155-176, 1996.

[2] B. Schulmeister and F. Wysotzki. Dipol - a hybrid piecewise linear classifier. *Machine Learning and Statistics: the Interface*, 133-151, New York, John Wiley and Sons, Inc, 1997.

[3] B.L. Lu and M. Ito. Task decomposition and module combination based on class relations: a modular neural network for pattern classification. *IEEE Transaction on Neural Networks*, 10(5), 1999.

[4] T.K. Kim and J. Kittler. Locally Linear Discriminant Analysis for Multimodally Distributed Classes for Face Recognition with a Single Model Image. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(3):318-327, 2005.

[5] M.L. Zhu and A.M. Martinez. Subclass Discriminant Analysis. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 28(8):1274-1286, 2006.

[6] J.J Wu, H. Hui, W. Peng, and J. Chen. Local Decomposition for Rare Class Analysis. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 814-823, 2007.

[7] P. Geibel, U. Brefeld, and F. Wysotzki. Perceptron and SVM learning with generalized cost models. In *Journal of Intelligent Data Analysis*, 8(5):439-455, 2004.

[8] B.L. Lu, K.A. Wang, M. Utiyama, and H. Isahara. A part-versus-part method for massively parallel training of support vector machines. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 1:735-740, 2004.

[9] H.B. Cheng, P.N. Tang, and R. Jin. Localized Support Vector Machine and Its Efficient Algorithm. In *Proceedings of the Seventh SIAM International Conference on Data Mining*, 2007.

[10] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *The Annals of Statistics*, 26(1):451-471, 1998.

[11] T.F. Wu, C.J. Lin, and R.C. Weng. Probability Estimates for Multi-class Classification by Pairwise Coupling. In *The Journal of Machine Learning Research*, 975-1005, 2004.

[12] J.H. Friedman. On Bias, Variance, 0/1-Loss, and the Curse-of-Dimensionality. In *Journal of Data Mining and Knowledge Discovery*, 1(1):55-77, 1997.

[13] C. Fraley, A.E. Raftery. Model-based clustering, discriminant analysis, and density estimation. In *Journal of the American Statistical Association*, 611-631, 2002.

[14] S. Bashir and E.M. Carter. High breakdown mixture discriminant analysis. In *Journal of Multivariate Analysis*, 93:102-111, 2005.

[15] S. Theodoridis and M. Mavroforakis. Reduced Convex Hulls: A Geometric Approach to Support Vector Machines. In *Signal Processing Magazine, IEEE*, 24(3):119-122, 2007.

[16] V.N. Vapnik. The nature of statistical learning theory. Springer-Verlag, New York, 1995.

[17] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *Springer*, 2001.

[18] F. Aurenhammer. Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. *Journal of ACM Computing Surveys*, 23:345-405, 1991.

[19] D. Newman, S. Hettich, C. Blake, and C. Merz. Uci repository of machine learning databases, 1998.

[20] C.C. Chang and C.J. Lin. LIBSVM : a library for support vector machines, 2001.

[21] P. Brazdil and J. Gama. Statlog datasets. http://www.liacc.up.pt/ML/statlog/datasets.html.

[22] R.M. Neal. Delve datasets. http://www.cs.utoronto.ca/ delve/data/datasets.html.